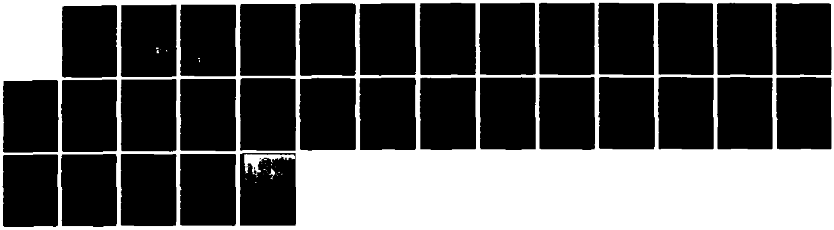
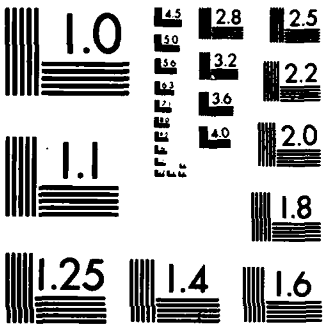
AD-A133 070 FACTORS AFFECTING SPECIFICATIONS FOR COMPUTER SOFTWARE 1/1
PROGRAMS(U) PERFORMANCE MEASUREMENT ASSOCIATES INC
VIENNA VA E M CONNELLY JUL 83 PMA-83-821
UNCLASSIFIED N00014-82-C-0499 F/G 9/2 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3

(12)

AD-A133070

PMA

This document has been approved for public release and sale; its distribution is unlimited.

83 09 29 018

12

ANNUAL
SUMMARY
REPORT

Edward M. Connelly

FACTORS AFFECTING SPECIFICATIONS FOR COMPUTER SOFTWARE PROGRAMS

REPORT NUMBER
83-821

This research is supported
by Engineering Psychology
Group, Office of Naval Research

July 1983

PMA
410 Pine Street, S.E.
Suite 300
Vienna, Virginia 22180
(703) 938-1600

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 83-821	2. GOVT ACCESSION NO. A133070	3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

Factors affecting specifications for computer software programs

5. TYPE OF REPORT & PERIOD COVERED

Annual Summary Report
1 July 82 - 30 June 83

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

Edward M. Connelly

8. CONTRACT OR GRANT NUMBER(s)

N00014-82-C-0499

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Performance Measurement Associates, Inc.
410 Pine Street, S.E. Suite 300
Vienna, VA 22180

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

61153N 42; RR04209;
RR0420901; NR 196-175

11. CONTROLLING OFFICE NAME AND ADDRESS

Engineering Psychology Group
Office of Naval Research
800 N. Quincy Street
Arlington, VA 22217

12. REPORT DATE

July 1983

13. NUMBER OF PAGES

29

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

Same

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, Distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Software Development, Requirement Specification, Automatic Processor

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The vulnerability of the software process development to errors or omissions in the requirements specification is well known. The goal of this program is to develop user aids that will help a neophyte user to work with an experienced software analyst in developing requirements specifications. The initial experiment, completed this year, is an investigation of the utility of cost aids as a function of task complexity.

The neophyte user developed various designs and requested cost information on previous designs, as well as, an automatic analysis of previous designs to reveal the most cost effective design and suggest new designs.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The vulnerability of the software process development to errors or omissions in the requirements specification is well known. The goal of this program is to develop user aids that will help a neophyte user to work with an experienced software analyst in developing requirements specifications. The initial experiment, completed this year, is an investigation of the utility of cost aids as a function of task complexity. The neophyte user developed various designs and requested cost information for those designs which were presented in various forms. The aids permitted the user to recall detailed information on previous designs, as well as, an automatic analysis of previous designs to reveal the most cost effective design and suggest new designs.

INTRODUCTION

Although errors in software occur in all stages of the software life cycle, errors that occur early in the cycle, especially in the preparation of software requirement specifications, are critical because they are difficult to detect. Also, the cost of correcting a requirement specification error increases as its detection is delayed through subsequent software development states. A further complication is, according to Boehm (1976) who voices a commonly stated observation, that most errors in software development occur in the early stages and frequently are errors of omission.

An incomplete or inaccurate requirement specification (RS) causes difficulty in other software development steps: systematic top-down design suffers from the lack of an incomplete top (specification), testing is inadequate due to lack of a complete, accurate requirement to test against, and project management suffers for the lack of a complete statement against which progress can be measured.

Existing methodologies, designed to improve software quality, unfortunately apply only to the design (preliminary and detailed design) and subsequent steps. These methodologies are directed to improving the steps in the software development process that

occur after the requirements have been specified and therefore neglect the process of producing the RS. The imbalance in the distribution of software methodologies and aids, which favor software design and test over preparation of RS, is illustrated by the lack of a requirement preparation "block" in typical descriptions of the software development process as shown in Figure 1. The process is typically visualized as beginning with a requirements block - as if to indicate that the process starts with the (supposedly existing) RS. Improvements in the requirements development process have been generally limited to development of notational methods for recording the requirements. Actually, of course, the process starts, as shown in Figure 2, with the user needs, which may or may not be well understood by the user, from which the RS must be developed. It is this process, the transformation of vague user needs into precise RS, often with the user working with a software expert, that is of interest here.

Background

At present, the tools available for software development fall into one of two classes: one class included the well-known design aids such as Structured Design (Yourdon & Constantine, 1979), Jackson's Method (Jackson, 1975), and Logical Construction of Programs (Warnier,

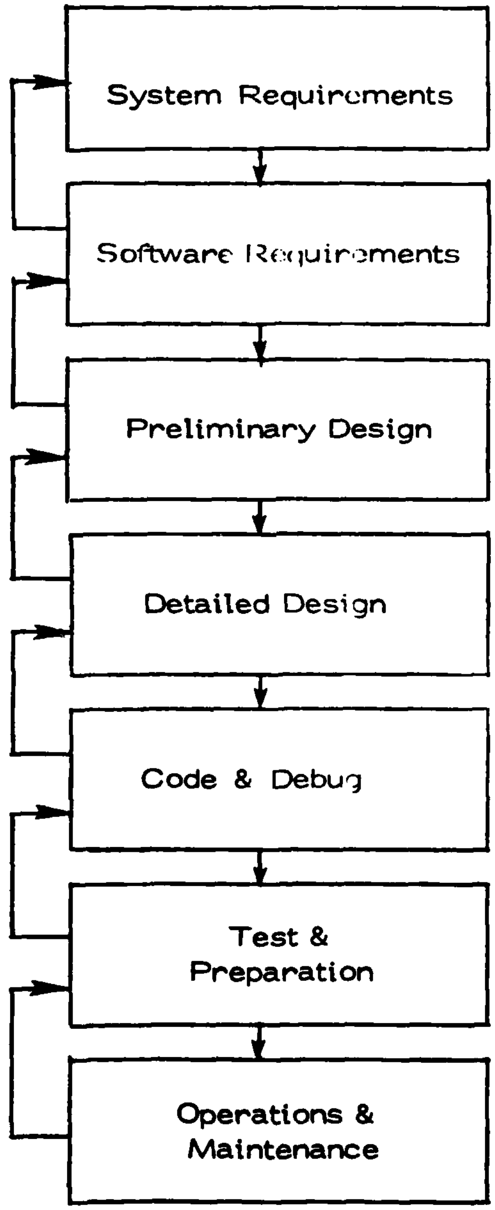


Figure 1. Typically [illegible] Erroneously Assumed Software Life Cycle

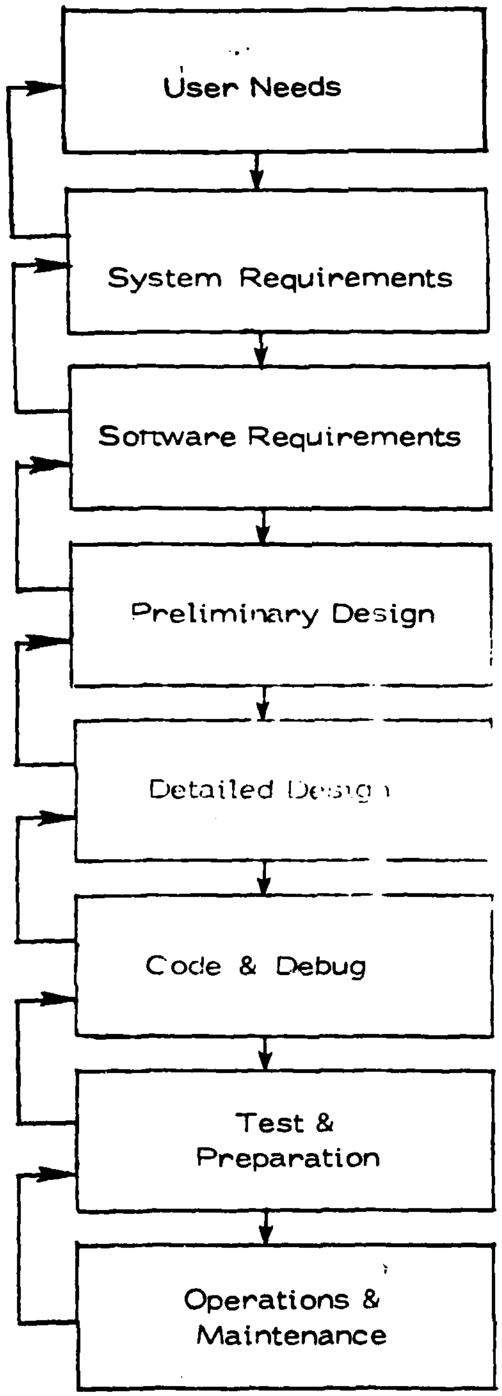


Figure 2. Actual Software Life Cycle

1974, and Orr, 1977). These are design aids that use RS, which are assumed to be correct, as a starting point.

The other class of aids provides a sturcture which can be used to record and analyze the RS. In this latter class is a system called ISDOS (Teichroew & Hershey, 1979) which used a problem statement language (PSL) and a problem statement analysis (PSA). ISDOS permits a formal description of the system in terms of entities, classes, and relationships, and automatically provides summaries such as problem statements, directories, hierarchical structure reports, graphical summaries of data flow and relationships. Another example in the class is a method called Software Requirements Engineering Programs (SREP) described by Boehm (1976) in a survey of methodologies. SREP uses the data management system of ISDOS and produces functional simulations from requirements statements. It is also used for configuration control, traceability from requirements to design, and report generation. A further method in the second class, Structured Analysis for Requirements Definition, is described by Ross and Schoman (1977). Part of that method is a Structured Analysis and Design Technique (SADT) for analyzing requirements using graphical techniques. All these methods

contain a common defect: that of providing a structure for recording the requirements, and then for analyzing those requirements but not for supporting the process of developing the RS from a user's needs.

In an extensive survey and review of the status of software requirements methods, Ramamoorthy & So (1978) identify the same requirements development problems referred to above; namely that a large percentage of the total errors in software development occur in the requirements specification, and that these errors cause serious problems leading to high costs, unresponsive products, slippage of production schedule, and difficulty in system operation and maintenance. Further, they briefly describe a number of methodologies for RS documentation and for aiding the software design process.

Research on Development of Requirements Specifications

Miller (1978) investigated the interactive process between a user (client) and software designer in analyzing the user's needs, establishing RS, and developing a software design. His description of the process uses four steps, and is presented below. For ease of reference in what follows we give the steps described by Miller even though interest here is limited to the first two steps.

Miller's four steps are:

1. Problem understanding, arriving at a general agreement as to what are:
   a. the goal objectives,
   b. the system or environments involved,
   c. constraints (on performance delivery costs, etc.),
   d. the resources available for system design development.
2. Functional requirement specification determining precisely what the final product must be like including:
   a. every important aspect of its internal performance,
   b. the characteristics of its embedded operator/user population,
   c. relationship to other systems and environments, and
   d. development constraints
3. Overall high level design translating the functional requirements into a comprehensive design which specifies the major components of the to-be developed product, and, for each describes:
   a. the goals to be achieved by the component,
   b. the characteristics of all factors to which the component is to be sensitive, i.e., the input,

c. the characteristics of the effects the component must achieve, i.e., the output,

d. the internal structures of the component, i.e., internal structures, and

e. the general principle of any operation sequences within the component information processing procedures.

4. Detailed design suitable for prototype development.

The steps of interest here are the first two steps which start with the initial discussions of the problem with the client and end with preparation of a formal RS. We do not treat the high level or detailed design steps.

Miller investigated the nature of the process of transforming the client's vague initial specification into a formal specification. He describes, in particular, the function of the client and software designer by describing the interchange between the two, and he suggests that the designers often use the technique of suggesting particular pieces of equipment or procedures that might be (or at least approximate) an acceptable solution. The client, in rejecting some of these suggestions will modify his own requirement statements.

As a result of this interchange the client will clarify his own understanding of the problem and they will mutually arrive at an acceptable solution.

Miller points out that the role of the designer is to provide facts about the real world in terms of properties of equipment and alternative solutions, as well as to ask questions which, while providing clarification, frequently may have the effect of inducing the client to identify a new problem or a better conceptualization of the present problem. He further identifies a sequence of six states which the client and designer use sequentially. The six states are:

1. Goal statement
2. Goal elaboration
3. (Sub) Solution outline
4. (Sub) Solution elaboration
5. (Sub) Solution explication
6. Agreement on (Sub) solution.

Miller indicates that this state sequence is used iteratively, but that sometimes the sequence is truncated to start a new one to pursue a different solution.

The results by Miller suggest that the process of transforming user's needs into a formal statement of requirements may benefit from the interchange between a client knowledgeable about his own needs and a software designer knowledgeable about the capabilities of computer systems. According to this model, the client's concept of his needs grows as a result of the interchange and he/she becomes aware of new and different solutions to his/her problem. New solutions evolve iteratively into one mutually accepted by both the client and designer as being complete and feasible.

The question then arises as to the need for an interchange to evolve the feasible solution. For instance, when preparing RS for a large computer system, it may not be possible for all the user-clients to have a useful interchange with one or more designers. Not only would these be multiple client-designer interchanges, but, there would be multiple interchanges (discussions of tradeoffs among the many interests) among the user-clients, and perhaps multiple interchanges among several designers. At present, when specifications for development of a large software system are considered, the user-client develops formal specifications without extensive interchange concerning the ultimate designs. The RS are then presented to designers - perhaps in the form of a request-for-quotation

RFQ). Such a procedure which is often used for large software projects is formal and prohibits the informal interchange of the type described above that might be used to advantage for a small software system.

METHOD OF APPROACH

In order to investigate the process by which an individual not skilled in the art of software would develop software RS, a series of four experiments are planned. Each experiment involves the design and evaluation of one or more aids. These are:

1. Aids to assist in finding the minimum cost system.
2. Aids to assist in generating complete specifications.
3. Aids to assist in building a vocabulary.
4. Aids to assist the neophyte users.

Experiment Task: An Inventory Control Problem

The experiment task is the development of a software specification for an inventory control problem by means of a recorded interaction between a user and a software designer.

Problem: Inventory Control

An important aspect of inventory control is the maintenance of records of present stock, amount of stock on order, recent transactions, and transaction histories. At periodic intervals, daily, weekly, or monthly, the old master file is read, transactions recorded, new stock levels computed, new stock order recommendations and other reports produced, and a new updated, master file produced. Specifications for a computer program to provide the

ADP inventory control would include specification of acceptable transactions, rules for entering new data or deleting old data, rules for computing stock levels, rules for computing purchase recommendations and computing stock history functions, rules for outputting reports and a new master file.

The particular inventory control problem considered for illustration of the research method is the processing of the old master file with current transactions to produce a new master file. The old and new master files are on magnetic tape. The current transaction file is in main memory.

Various tradeoff's involving data accuracy and data protection, as well as speed of the process, must also be considered in preparing the software specifications. Software specifications do not delineate the way a particular feature is to be implemented; specifications state the combination of features required in the software product. Consequently, the specification can consist of <u>logical</u> functions which indicate that combination of features.

A score indicating the quality of the RS is the number of factors that are included in the specification that either were not included in the problem statement or should not be included in the specification because of the high cost required to implement them.

Participants

(Two participants in each experiment trial)

- One participant, not an experienced analyst/programmer, but experienced in a particular subject-matter field, is the "user."
- Another participant, an experienced analyst/programmer, is the "software designer."

Procedure

The participants are given instructions individually via video tape presentations. These instructions include procedures, goals, and the method of scoring to be used in evaluating the experiment. Following the presentation, a written inventory control problem statement is given to the user. The problem statement identifies the various functions that could be included in the specification and the relative importance of each function. Further, the user is told that the present objective is to interact with a designer to jointly develop a software specification for the problem. The user is instructed that all necessary features (identified as necessary in the problem statement) of the process plus additional features (that are also identified in the problem statement) that are achievable at low cost should be included in the system specification. The participant is also told that additional

factors which may result in programming delays or otherwise incur a large additional cost are not to be included in the specification. Thus the user's task is: to understand the problem, to establish the communication with the designer, to work with the designer to determine what features must be included in the specification and what additional features can be included in low cost, and to insure that the necessary functional relationships among problem features are included in the software specification.

The designer is also given instructions individually via video tape. He/she is given the basic vocubulary and information on the capability of the operating system to be used for the software system. He/she is instructed to establish a communication with a user, offer information regarding feasibility and relative cost (programming effort and program efficiency) of various features. Finally, he/she is instructed to prepare a software specification for the process.

After a short period of time in which the user reviews the problem statement, the user and designer are permitted to communicate via a keyboard link (computer controlled) to exchange information concerning the problem and to arrive at a solution in the form of the software specification.

Progress to Date

For the first experiment identified above, aids were designed to facilitate specification of a minimum cost system. The participants interacted with a simulated software expert which provided cost information feedback for each of the participant's trial specifications. The procedure used to find the requirements specification for the least-cost system is as follows: The participants input a trial RS and received as feedback the cost of that system. Next he/she reviewed the specification and the associated system cost along with any other specifications (and their system costs) previously input. Based on that review, the RS were modified in an attempt to specify a lower cost system. This is an iterative, "cut and try" procedure in which the user employed knowledge of the cost of previous specifications and knowledge of the function of system parts to develop the RS for the next iteration.

This iterative process continued until the total allowed session time was used (1 hour) or the user believed that the requirements for the minimum cost system had been specified.

Three levels of cost feedback aids were used. The base level cost feedback configuration, where a minimum of cost processing and feedback was provided, gave the user information only on the

total cost for the specification. The next level of aid processing provided cost information for each part of the system as well as the total cost. And finally, at the highest level, users were given the information from the first two levels plus an additional analysis showing the relationship between cost and the elements of the RS previously entered.

The data has been collected for this first experiment and the data analyses is now in progress.

Future Plans

In addition to the data analysis on the first experiment, work has been started on vocabulary building aids in which two individuals will work together by inputting data on separate but communicating terminals to first build a working vocabulary and then to develop a requirements specification. The aids will assist the two participants to rapidly build efficient vocabularies for the requirements specification. Following completion of the vocabulary building experiment, the aids for generating complete (and accurate) RS and aids to assist the neophyte user will be developed and tested.

REFERENCES

Boehm, B. W. Software engineering. IEEE Transactions on Computers, December 1976, vol C-25, No. 12, pp. 1226-41.

Jackson, M. A. Principles of program design. Academic Press, New York, 1975.

Miller, L. A. Behavioral studies of the programming process. Final Report for ONR, Work Unit NR197-020, Code 455 October 1978.

Orr, K. & Associates, Inc. Structured requirements definition. Topeka, Kansas 1981.

Ramamoorthy, C. V. & So, H. H. Software requirements and specifications: status and perspectives. Presented at the IEEE Computer Society's Second International Computer Software and Applications Conference, Chicago, Illinois, 1978.

Ross, D. T. & Schoman, D. E. Jr. Structured analysis for requirements definition. IEEE Transactions on Software Engineering, January 1977.

Teichroew, D. & Hershey, E. A. III. PSL/PSA: a computer-aided technique for structured documentation and analysis of information processing systems. Presented at the IEEE Computer Society's Third International Computer Software & Applications Conference, Chicago, Illinois 1979.

Warnier, J. D. The logical construction of programs. 3rd ed, trans. B. M. Flanagan, Van Nostrand Reinhold, New York 1976.

Yourdon, E. & Constantine, L. L. Structured design: fundamentals of a discipline of computer program and systems design. Prentice-Hall, Inc., New Jersey, 1979.

DISTRIBUTION LIST

OSD

CAPT Paul R. Chatelier
Office of the Deputy Under Secretary of Defense
OUSDRE (E& LS)
Pentagon, Room 3D129
Washington, D.C. 20301

Dr. Dennis Leedom
Office of the Deputy Under Secretary of Defense (C^3I)
Pentagon
Washington, D.C. 20301

Department of the Navy

Engineering Psychology Group
Office of Naval Research
Code 442 EP
Arlington, VA 22217 (2 cys.)

Aviation & Aerospace Technology Programs
Code 210
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Communication & Computer Technology Programs
Code 240
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Tactical Development & Evaluation Support Programs
Code 230
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Department of the Navy

Manpower, Personnel & Training Programs
Code 270
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Information Sciences Division
Code 433
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Special Assistant for Marine Corps Matters
Code 100M
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

CDR James Offutt, Officer-in-Charge
ONR Detachment
1030 East Green Street
Pasadena, CA 91106

Director
Naval Research Laboratory
Technical Information Division
Code 2627
Washington, D.C. 20375

Dr. Michael Melich
Communications Sciences Division
Code 7500
Naval Research Laboratory
Washington, D.C. 20375

Department of the Navy

Dr. J. S. Lawson
Naval Electronic Systems Command
NELEX-06T
Washington, D.C. 20360

Dr. Robert G. Smith
Office of the Chief of Naval
Operations, OP987H
Personnel Logistics Plans
Washington, D.C. 20350

Combat Control Systems Department
Code 35
Naval Underwater Systems Center
Newport, RI 02840

Human Factors Department
Code N-71
Naval Training Equipment Center
Orlando, FL 32813

Dr. Alfred F. Smode
Training Analysis and Evaluation
Group
Orlando, FL 32813

CDR Norman E. Lane
Code N-7A
Naval Training Equipment Center
Orlando, FL 32813

Dr. Gary Poock
Operations Research Department
Naval Postgraduate School
Monterey, CA 93940

Dean of Research Administration
Naval Postgraduate School
Monterey, CA 93940

Department of the Navy

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, D.C. 20380

Dr. L. Chmura
Naval Research Laboratory
Code 7592
Computer Sciences & Systems
Washington, D.C. 20375

HQS, U.S. Marine Corps
ATTN: CCA40 (Major Pennell)
Washington, D.C. 20380

Commanding Officer
MCTSSA
Marine Corps Base
Camp Pendleton, CA 92055

Chief, C^3 Division
Development Center
MCDEC
Quantico, VA 22134

Human Factors Technology Administrator
Office of Naval Technology
Code MAT 0722
800 North Quincy Street
Arlington, VA 22217

Commander
Naval Air System Command
Human Factors Programs
NAVAIR 334A
Washington, D.C. 20361

Department of the Navy

Commander
Naval Air Systems Command
Crew Station Design
NAVAIR 5313
Washington, D.C. 20361

Mr. Philip Andrews
Naval Sea Systems Command
NAVSEA 03416
Washington, D.C. 20362

Commander
Naval Electronics Systems Command
Human Factors Engineering Branch
Code 81323
Washington, D.C. 20360

Larry Olmstead
Naval Surface Weapons Center
NSWC/DL
Code N-32
Dahlgren, VA 22448

Dr. George Moeller
Human Factors Engineering Branch
Submarine Medical Research Lab
Naval Submarine Base
Groton, CT 06340

Commander, Naval Air Force,
U.S. Pacific Fleet
ATTN: Dr. James McGrath
Naval Air Station, North Island
San Diego, CA 92135

Navy Personnel Research and
Development Center
Planning & Appraisal Division
San Diego, CA 92152

Department of the Navy

Dr. Robert Blanchard
Navy Personnel Research and
Development Center
Command and Support Systems
(Code 17)
San Diego, CA 92152

CDR J. Funaro
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

Mr. Stephen Merriman
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

Mr. Jeffrey Grossman
Human Factors Branch
Code 3152
Naval Weapons Center
China Lake, CA 93555

Human Factors Engineering Branch
Code 1226
Pacific Missile Test Center
Point Mugu, CA 93042

Dean of the Academic Departments
U.S. Naval Academy
Annapolis, MD 21402

Dr. S. Schiflett
Human Factors Section
Systems Engineering Test Directorate
U.S. Naval Air Test Center
Patuxent River, MD 20670

Department of the Navy

Mr. Harry Crisp
Code N 51
Combat Systems Department
Naval Surface Weapons Center
Dahlgren, VA 22448

CDR C. Hutchins
Code 55
Naval Postgraduate School
Monterey, CA 93940

Mr. J. Impagliazzo
Code 101
Naval Underwater Systems Center
Newport, RI 02840

Department of the Army

Mr. J. Barber
HQS, Department of the Army
DAPE-MBR
Washington, D.C 20310

Dr. Edgar M. Johnson
Technical Director
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Director, Organizations and
Systems Research Laboratory
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Technical Director
U.S. Army Human Engineering Labs
Aberdeen Proving Ground, MD 21005

Department of the Air Force

U.S. Air Force Office of Scientific
Research
Life Sciences Directorate, NL
Bolling Air Force Base
Washington, D.C. 20332

AFHRL/LRS TDC
Attn: Susan Ewing
Wright-Patterson AFB, OH 45433

Chief, Systems Engineering Branch
Human Engineering Division
USAF AMRL/HES
Wright-Patterson AFB, OH 45433

Dr. Earl Alluisi
Chief Scientist
AFHRL/CCN
Brooks Air Force Base, TX 78235

Foreign Addressees

Director, Human Factors Wing
Defense & Civil Institute of
Environmental Medicine
Post Office Box 2000
Downsview, Ontario M3M 3B9
Canada

Dr. A. D. Baddeley
Director, Applied Psychology Unit
Medical Research Council
15 Chaucer Road
Cambridge, CB2 2EF England

Other Government Agencies

Defense Technical Information Center
Cameron Station, Bldg. 5
Alexandria, VA 22314 (12 copies)

Other Government Agencies

Director, System Sciences Office
Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

Dr. M. Montemerlo
Human Factors & Simulation Technology, RTE-6
NASA HQS
Washington, D.C. 20546

Other Organizations

Dr. Jesse Orlansky
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

Dr. Robert T. Hennessy
NAS - National Research Council (COHF)
2101 Constitution Avenue, N.W.
Washington, D.C. 20418

Dr. Robert C. Williges
Department of Industrial Engineering and or
Virginia Polytechnic Institute and State University
130 Whittemore Hall
Blacksburg, VA 24061

Dr. Deborah Boehm-Davis
General Electric Company
Data & Information Systems
1755 Jefferson Davis Highway
Arlington, VA 22202

Other Organizations

Dr. Paul E. Lehner
PAR Technology Corp
P.O. Box 2005
Reston, VA 22090

Dr. James O. Chinnis
Decision Science Consortium
Suite 721
7700 Leesburg Pike
Falls Church, VA 22043

Dr. Alan C. Morse
Intelligent Software Systems Inc.
529 Belchertown Road
Amherst, MA 01002

Dr. Richard Pew
Bolt Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02238

END

FILMED

10-83

DTIC